Problem Statement

Task

Run a http server inside a docker container in an EC2 server which returns a list of all EC2s running in a region in an AWS account along with their corresponding IPs and Tags when called on the `/ec2` path

The stack should be as IaC(Infrastructure as Code) and should have the following components:

EC2

Launch template

ASG

Target Group

Entry in the ALB

Route 53 entry

Security Group

Inside the EC2, run your application inside a docker container and expose the application on port 8000

Good documentation would go a long way to show us how you approached this task with your techno-magic.

Brownie points

Make a terraform module or CDK construct which takes values like EC2 instance type, Volume size, AMI Name, Number of EC2s, DNS name

Using Ansible or any config management tool to install docker and other dependencies in the EC2

Solution:-

Resources used:

- AWS as a cloud provider
- Terraform for infrastructure as code
- Docker for conterization

In terraform code:-

- **autoscaling.tf**

It has two resources

aws_launch_configuration :- this is used to for giving the configuration of ec2 instances that will be created by aws

aws_autoscaling_group: This will actually create the autoscaling group with min_size as 1 and max_size as 2

- **key.tf**

The basic use of this file is to create the public and private keys and download them in local which can be used to perform ssh on the ec2 instances created by autoscaling group

- **lb.tf**

It has four resources inside it

1. aws_lb :- this resource is use to create the load balancer
2. aws_lb_target_group: this resource is use to create the target groups, with the health check which will be our ASG basically
3. aws_lb_listener: this resource describe on which port ec2 instances inside alb will listen to target, this will also create a default forward listener rule
4. aws_lb_listener_rule:- since we want to hit the value /ec2 with load balancer url we need to create the redirect rule in alb which is for redirecting to port 8000

route53.tf

It has two resources

1. aws_route53_zone :-it is use to create the hosted zone in route53
2. aws_route53_record: this will create a A record with the alb created above

The output of this terraform will give us the nameserver records which can be used to in any domain name service to point that domain name to our load balancer url

SG.tf

This resource is use to create security group which will be use to communication bw load balancer ec2 and internet

Two security group gets created:

1. Instance:
    a. ssh port(22) open from anywhere
    b. port 8000 and 80 open from LB
2. ELB :-
    a. 80 and 8000 open from anywhere

So we will able to access the website from LB url only

Variables.tf

It contain all the variables req in this terraform code to apply

Versions.tf

It contains the provider block and required terraform version to work for this task

install-httpd.sh

This is the script which is used by all the instances created by autoscaling group to configure everything

Steps:

- Installs docker
- Create a docker file
- Build the image
- Deploy the container

Dockerfile:-

The file to authenticate container to AWS and and run different command to describe instances and use them to list running instances there private ips and tags of ec2 instance

Final result:-

The httpd server is hosted in apache and when i hit the url of my LB or domain i get the list of running instances along with the private ips and tags attached to that instance

Screenshot for above commitment

Instanceid=["i-0f8815a53505d3a8b"] PrivateIP=["172.31.2.13"] TAGS=[[{ "Key": "Name", "Value": "ec2 instance" }, { "Key": "aws:autoscaling:groupName", "Value": "task-autoscaling" }]]